SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Matej Vitasović

Zagreb, 2014.

SVEUČILIŠTE U ZAGREBU

FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

## Statička neuronska mreža s unipolarnom sigmoidalnom i Gaussovom aktivacijskom funkcijom

Mentor: Student:

Doc.dr.sc. Danko Brezak Matej Vitasović

Zagreb, 2014.

*Pod punom moralnom odgovornošću izjavljujem da sam samostalno izradio ovaj rad, koristeći znanja stečena tijekom studija i navedenu literaturu.*

*Matej Vitasović*

SVEUČILIŠTE U ZAGREBU
**FAKULTET STROJARSTVA I BRODOGRADNJE**
Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo materijala i mehatronika i robotika

| Sveučilište u Zagreb<br>Fakultet strojarstva i brodogradnje | |
|---|---|
| Datum 18-09-2014 | Prilog |
| Klasa: 602-04/14-6/2 | |
| Ur.broj: 15-1703-14-318 | |

# ZAVRŠNI ZADATAK

Student: **Matej Vitasović** Mat. br.: 0035183664

Naslov rada na hrvatskom jeziku: **Statička neuronska mreža s unipolarnom sigmoidalnom i Gaussovom aktivacijskom funkcijom**

Naslov rada na engleskom jeziku: **Static neural network with unipolar sigmoid and Gaussian activation function**

Opis zadatka:

Za aktivacijsku funkciju neurona skrivenog sloja kod statičkih neuronskih mreža najčešće se odabire sigmoidalna funkcija s fiksnim nagibom. Među ostalim potencijalnim aktivacijskim funkcijama interesantnom se čini primjena Gaussove radijalne bazne funkcije, koja omogućava smanjenje strukture skrivenog sloja mreže izostavljanjem tzv. bias neurona.
Stoga je u ovome radu analizirana primjena unipolarne sigmoidalne aktivacijske funkcije sa i bez adaptacije nagiba, te Gaussove funkcije s adaptabilnim parametrima na nizu standardnih problema.

U radu je potrebno:

1. Izraditi programsku podršku za tri varijante statičke unaprijedne neuronske mreže učene EBP metodom učenja, koje imaju unipolarnu sigmoidalnu aktivacijsku funkciju sa i bez adaptacije nagiba, te Gaussovu funkciju s adaptacijom pozicije i širine.
2. Generirati odzive mreže za sve varijante mreže primjenom nekoliko standardnih usporednih testova.
3. Usporediti dobivene rezultate različitih struktura mreže.
4. Izvesti zaključke rada.

Zadatak zadan:
11. studenog 2013.

Rok predaje rada:
**1. rok:** 21. veljače 2014.
**2. rok:** 12. rujna 2014.

Predviđeni datumi obrane:
**1. rok:** 3., 4. i 5. ožujka 2014.
**2. rok:** 22., 23. i 24. rujna 2014.

Zadatak zadao:

Doc. dr. sc. Danko Brezak

Predsjednik Povjerenstva:

Prof. dr. sc. Zoran Kunica

Sadržaj

# Sažetak

Za aktivacijsku funkciju neurona skrivenog sloja kod statičkih neuronskih mreža najčešće se odabire sigmoidalna funkcija s fiksnim nagibom. Među ostalim potencijalnim aktivacijskim funkcijama interesantnom se čini primjena Gaussove radijalne bazne funkcije, koja omogućava smanjenje strukture skrivenog sloja mreže izostavljanjem tzv. *Bias* neurona.

Stoga je u ovome radu analizirana primjena unipolarne sigmoidalne aktivacijske funkcije sa i bez adaptacije nagiba, te Gaussove funkcije s adaptibilnim parametrima na nizu standardnih problema.

**Ključne riječi:** Neuronske mreže, unipolarna sigmoidalna funkcija, Gaussova radijalna bazna funkcija, parametri učenja neuronske mreže

# Popis slika

# Popis tablica

# Popis oznaka

| | |
|---|---|
| $a$ | parametar kaotičnog dinamičkog sustava |
| $b$ | parametar kaotičnog dinamičkog sustava |
| $Bias$ | neuron bez ulaza sa konstantnim izlazom 1 |
| $C$ | nagib unipolarne sigmoidalne aktivacijske funkcije sakrivenog sloja |
| $d_k$ | željena izlazna vrijednost *k*-tog izlaza neuronske mreže |
| $\bar{d}$ | srednja vrijednost izlazne datoteke učenja |
| $D$ | izlazni skup datoteka za učenje |
| $\boldsymbol{e}$ | vektor pogreške |
| $E$ | suma kvadrata pogreške (funkcija cilja) |
| $I$ | broj neurona ulaznog sloja mreže |
| $J$ | broj neurona sakrivenog sloja mreže |
| $K$ | broj neurona izlaznog sloja mreže |
| $K_p$ | pojačanje aktivacijske funkcije izlaznog sloja |
| $N$ | broj parova ulazno-izlaznih redaka datoteke učenja |
| $NRMS$ | normalizirani korijen srednje kvadratne pogreške |
| $net$ | vrijednost funkcije sume |
| $net_j$ | vrijednost funkcije sume *j*-tog neurona |
| $net_{Hj}$ | vrijednost funkcije sume *j*-tog neurona sakrivenog sloja |
| $net_{Ok}$ | vrijednost funkcije sume k-tog neurona izlaznog sloja |
| $O_k$ | *k*- ti izlaz neuronske mreže |
| $\boldsymbol{O}$ | vektor (matrica) izlaza iz mreže |
| $T$ | vremenska konstanta [s] |
| $T_0$ | period uzorkovanja [s] |
| $t_j$ | parametar Gaussove funkcije (pozicija centra aktivacijske funkcije j-tog neurona skrivenog sloja) |
| $u(n)$ | trenutna vrijednost ulaza sustava |
| $v_{ji}$ | težinski koeficijent veze između *j*-tog neurona sakrivenog sloja te *i*-tog neurona ulaznog sloja |
| $\boldsymbol{V}$ | vektor (matrica) težinskih koeficijenata sakrivenog sloja |
| $w_{kj}$ | težinski koeficijent veze između *k*-tog neurona izlaznog sloja i *j*-tog neurona sakrivenog sloja |
| $\boldsymbol{W}$ | vektor (matrica) težinskih koeficijenata izlaznog sloja |
| $\boldsymbol{Y}$ | vektor (matrica) izlaznih vrijednosti neurona sakrivenog sloja |
| $y$ | vrijednost izlaza neurona |
| $Z_i$ | *i*-ti ulaz neuronske mreže |
| $\boldsymbol{Z}$ | ulazni skup datoteka za učenje |
| $\vartheta$ | parametar učenja |
| $\vartheta(n)$ | trenutna vrijednost parametra učenja |
| $\vartheta(n+1)$ | nova vrijednost parametra učenja |
| $\Delta\vartheta(n)$ | trenutna promjena parametra učenja |

| | |
|---|---|
| $\Delta$ | iznos promjena težina u jednom koraku |
| $\delta$ | parametar algoritma povratnog prostiranja pogreške |
| $\delta_{Ok}$ | parametar algoritma povratnog prostiranja pogreške izlaznog sloja |
| $\gamma$ | aktivacijska funkcija neurona |
| $\eta$ | koeficijent brzine učenja gradijentnog algoritma |
| $\sigma_j$ | Parametar Gaussove aktivacijske funkcije (širina aktivacijske funkcije *j*-tog neurona skrivenog sloja) |
| $\sigma_{d_n}$ | standardno odstupanje |
| $\nabla E$ | gradijent pogreške |
| $\tau$ | vremensko kašnjenje dinamičkog sustava |

# 1. Uvod

Izvorna ideja za umjetnu neuronsku mrežu produkt je težnji da se modelira biofiziologija mozga čovjeka, s ciljem razumijevanja i objašnjenja kako isti funkcionira. Model bi trebao biti sposoban imitirati funkciju ljudskog mozga, odnosno trebao bi biti sposoban obrađivati informacije na isti način na koji to ljudski mozak radi. Danas se neuronske mreže koriste u rješavanju široke lepeze problema prepoznavanja uzoraka , te predikcije, aproksimacije i interpolacije [1]. Njihova je struktura ovisna o vrsti problema i namjeni, pa ih tako možemo kategorizirati na više načina – time govorimo o vrsti neurona koji čini mrežu, aktivacijskoj funkciji neurona, broju slojeva mreže, načinu učenja mreže, te ovisnosti parametara mreže o vremenu.

Umjetna neuronska mreža obrađena u ovom radu spada u kategoriju statičkih mreža, što znači da niti jedan parametar mreže ne ovisi o vremenu. Dakle, ne uzimaju se u obzir efekti kašnjenja signala, kao i to da se na ulaze neurona ne dovode izlazi tog istog neurona, ili izlazi bilo kojeg drugog neurona. Doduše, trebamo napomenuti da se u ispitivanju mreže testom predikcije kaotičnog sustava, izlaz mreže spajao na ulazni sloj, no temelj takve mreže je još uvijek bio statički umjetni neuron, što ipak mrežu čini statičkom.

Nadalje, ova mreža je i unaprijedna, sa svojstvom da su izlazni signali neurona *n*-tog sloja spojeni isključivo na ulaze *(n+1)*-tog sloja, te mogu utjecati samo na taj sloj.

Mreža se, konačno, može opisati i kao višeslojna, s obzirom da posjeduje tri distinktivna sloja neurona – ulazni, skriveni, te izlazni sloj. Važno je napomenuti da se ulazni sloj često ne smatra slojem, te se niti ne broji u definiranju slojevitosti mreže, zbog činjenice da neuroni ulaznog sloja ne posjeduju funkciju sume i aktivacijsku funkciju, što su temeljne karakteristike umjetnih neurona. Tako, iako naša mreža posjeduje spomenuta tri sloja, ona se po definiciji naziva *dvoslojnom*.

Zadatak neuronske mreže je da se postupnom promjenom svojih parametara približi traženom izlazu sa svojim izlazom. Postupak promjene parametara mreže naziva se učenjem mreže i može se obaviti na više različitih načina. U ovom radu koristiti ću se algoritmom povratnog prostiranja pogreške (eng. *Error Backpropagation Algorithm*, u daljnjem tekstu 'EBP algoritam') , s obzirom da je to najpoznatiji i najčešće primjenjivani način promjene parametara učenja. EBP algoritam je iteracijske prirode, što znači da mreža mora obaviti određeni broj koraka prije nego što se njen izlaz može približiti traženom izlazu s dovoljnom, odnosno traženom točnošću. Istovremeno, algoritmom se pokušava aproksimirati traženi izlaz, a posljedica je toga da stvarni izlaz mreže nikada ne može dati traženi sa stopostotnom točnošću [1].

Važno svojstvo neurona je aktivacijska funkcija koju koristimo za preslikavanje izlaza funkcije sume (suma otežanih ulaza u neuron) na izlaz samog neurona. Obično se odabire monotono rastuća funkcija sa zasićenjem, no u posljednje se vrijeme pojavljuje i implementacija Gaussove funkcije. U ovom radu ću stoga analizirati primjenu unipolarne sigmoidalne funkcije sa i bez adaptacije nagibe, te Gaussove funkcije s adaptibilnim parametrima.

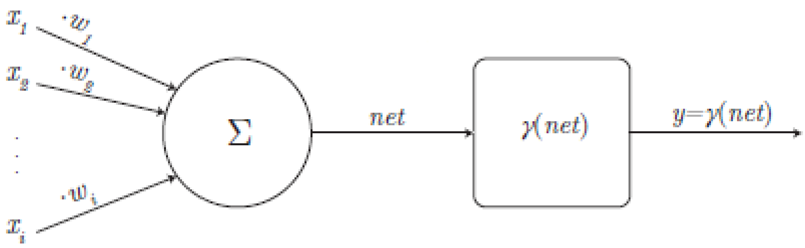
Problemi na kojima će mreže sa navedenim aktivacijskim funkcijama biti testirane i uspoređene su, u literaturi uvriježeni, standardni problemi: XOR problem, te predviđanje ponašanja kaotičnog nelinearnog sustava (Glass-Mackey)

U prvom djelu ovog rada opisati ću strukturu neuronske mreže – matematički model neurona, kao i model mreže. Govorit ću o propagaciji signala, te algoritmu promjene parametara učenja. Opisati ću aktivacijske funkcije uspoređivane u radu i njihova svojstva i utjecaj na strukturu mreže, konkretno, na potrebu korištenja *Bias* neurona kod unipolarnih sigmoidalnih aktivacijskih funkcija. Nadalje, opisati ću probleme na kojima testiram mrežu, te naposljetku i rezultate pojedine mreže i njihovu usporedbu.

# 2. Unaprijedna statička višeslojna neuronska mreža

## 2.1 Umjetni statički neuron

Kao što je spomenuto u uvodu, statička mreža je ona mreža čiji se parametri ne mijenjaju u ovisnosti o vremenu. Osnovni blok kojim se gradi statička mreža se naziva umjetni statički neuron. Njegova je struktura shematski prikazana slikom 2.1. [2].




**2.1. Umjetni statički neuron**

Statički se neuron još naziva i perceptron, te McCulloch-Pittsov neuron [1]. Tijelo neurona čine dva dijela, odnosno dvije funkcije – funkcija sume *net* , te aktivacijska funkcija $\gamma$, kao što je vidljivo na slici.

Funkcija sume *net* zbraja otežane ulaze u neuron, koji su ujedno i izlazi iz neurona koji pripadaju prethodnom sloju.

$$net_j = \sum_i w_{ji} \cdot x_i \tag{2.1}$$

Gdje indeks $i = 1,2, \dots, n$ označava ulaze u neuron (također, broj neurona prethodnog sloja, uračunat i *Bias,*kada je potrebno), dok indeks $j = 1,2, \dots, n$ označava neuron sloja na kojeg se funkcija sume odnosi.

O aktivacijskoj se funkciji nešto reklo u uvodu, te će se više reći i kasnije. Općenito, ona preslikava $net_j$ u izlaz neurona $y_j$.

$$y_j = \gamma(net_j) \tag{2.2}$$

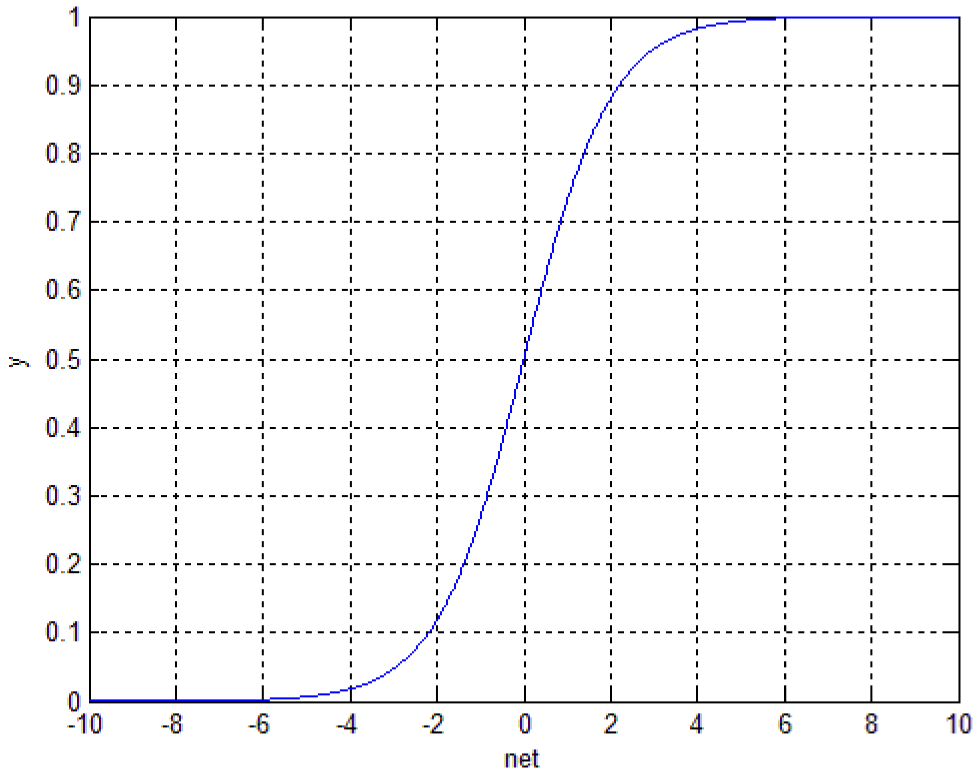
Gdje indeks $j = 1,2, \dots, n$ označava isto što i u funkciji sume. Aktivacijske funkcije koje ću primijeniti u radu biti će unipolarna funkcija sa i bez adaptacije nagiba, te Gaussova funkcija s adaptibilnim parametrima (širina i pozicija centra funkcije).

## 2.2 Unipolarna sigmoidalna funkcija

Nelinearna unipolarna sigmoidalna funkcija prikazana je slikom 2.2., te matematičkim izrazom u jednadžbi (2.3). Uz bipolarnu sigmoidalnu funkciju, ona je jedna od najčešće korištenih aktivacijskih funkcija.

$$y = \frac{1}{1 + e^{(-C \cdot net)}} \tag{2.3}$$

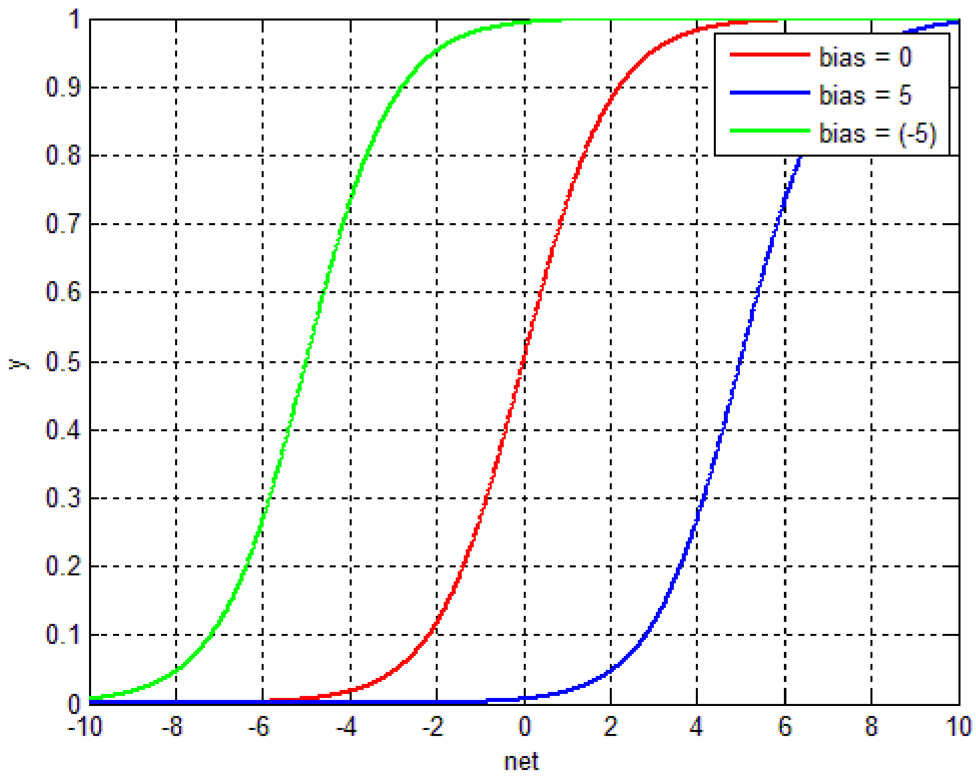
Za malene pozitivne i negativne vrijednosti ulaza, njen izlaz varira između nula i jedan, dok za sve veće pozitivne i negativne vrijednosti ulaza, ona daje izlaze veoma blizu nuli i jednici, drugim riječima, asimptotski se približava tim vrijednostima. U nekim slučajevima to je vrlo zgodna stvar, jer je izlaz automatski normiran između dviju vrlo često upotrebljavanih granica samim svojstvom funkcije. Također, još jedno svojstvo te funkcije je da je izlaz uvijek pozitivan, dakle izlaz funkcije ne može biti negativan.




**2.2. Unipolarna sigmoidalna funkcija**

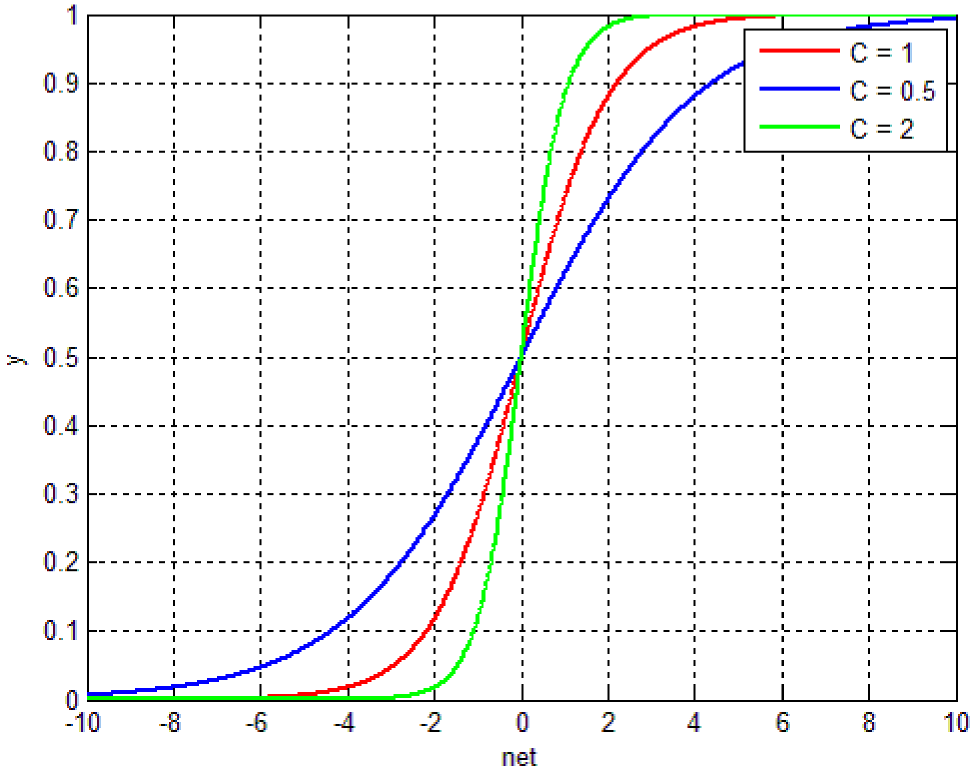
Neuronske mreže koje implementiraju unipolarnu sigmoidalnu funkciju kao aktivacijsku funkciju moraju za ulaz svakom neuronu svakog sloja (osim izlaznog) dodati jedinicu i pripadajuće težine. To se izvodi na način da se svakom sloju osim izlaznog dodaje jedan neuron čiji je izlaz uvijek jedinica. Takav se neuron u literaturi naziva *Bias* neuronom. Utjecaj *Bias*-a je pomak aktivacijske funkcije po apscisi, što je često nužnost za uspješno učenje neuronske mreže. Na slici 2.3. može se vidjeti utjecaj nekih proizvoljnih vrijednosti na unipolarnu sigmoidalnu funkciju. Utjecaj *Bias* neurona je analogan tome. Rezultat je dobiven jednadžbom (2.3), gdje je *bias* imao vrijednosti -5, 0 i 5, što je trebalo predstaviti izlaz *Bias* neurona pomnožen sa pripadajućim težinama, te je izoliran iz funkcije sume kako bi utjecaj bio očitiji u izrazu. Koeficijent nagiba $C$ je bio jednak jedinici.

$$y = \frac{1}{1 + e^{C\cdot(-net+bias)}} \tag{2.4}$$




**2.3. Grafovi unipolarne sigmoidalne funkcije za proizvoljne vrijednosti *bias* ulaza**

Još jedan važan parametar unipolarne sigmoidalne funkcije jest koeficijent nagiba $C$. U ovom radu ćemo, između ostaloga, usporediti utjecaj promjene parametra koeficijenta nagiba na sposobnost učenja neuronske mreže. Teoretski bi mogućnost adaptacije ubrzala konvergenciju ka rješenju zbog veće fleksibilnosti mreže koja može na takav način prilagođavati funkcije. Na slici 2.4. može se vidjeti utjecaj na oblik funkcije za proizvoljne vrijednosti koeficijenta.




**2.4. Grafovi unipolarne sigmoidalne funkcije za proizvoljne vrijednosti koeficijenta nagiba**

Prve derivacije funkcije po oba parametra potrebne su kako bi se kasnije mogao računati gradijent pogreške. U izrazima (2.5) i (2.6) može se vidjeti da su prve derivacije unipolarne sigmoidalne funkcije relativno jednostavne, što se može pridodati prednostima korištenja takve aktivacijske funkcije.

$$y'(net) = C \cdot y(1-y) \tag{2.5}$$

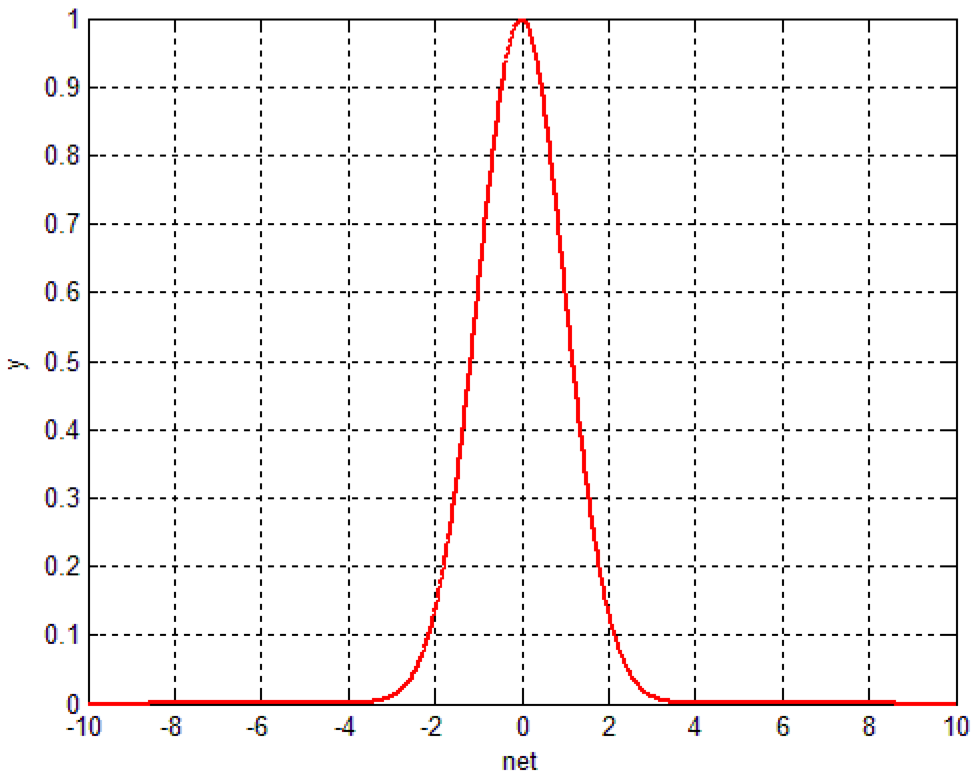
$$y'(C) = net \cdot y(1-y) \tag{2.6}$$

## 2.3 Gaussova funkcija

Još jedna aktivacijska funkcija koja će se testirati u ovom radu je Gaussova funkcija, prikazana na slici 2.5., te matematičkim izrazom u jednadžbi (2.7).

Gaussova funkcija ima karakterističan zvonolik oblik, te izlaz funkcije ovisi o udaljenosti ulaza od centra 'zvona'. Zbog toga Gaussova funkcija spada u kategoriju radijalnih funkcija.

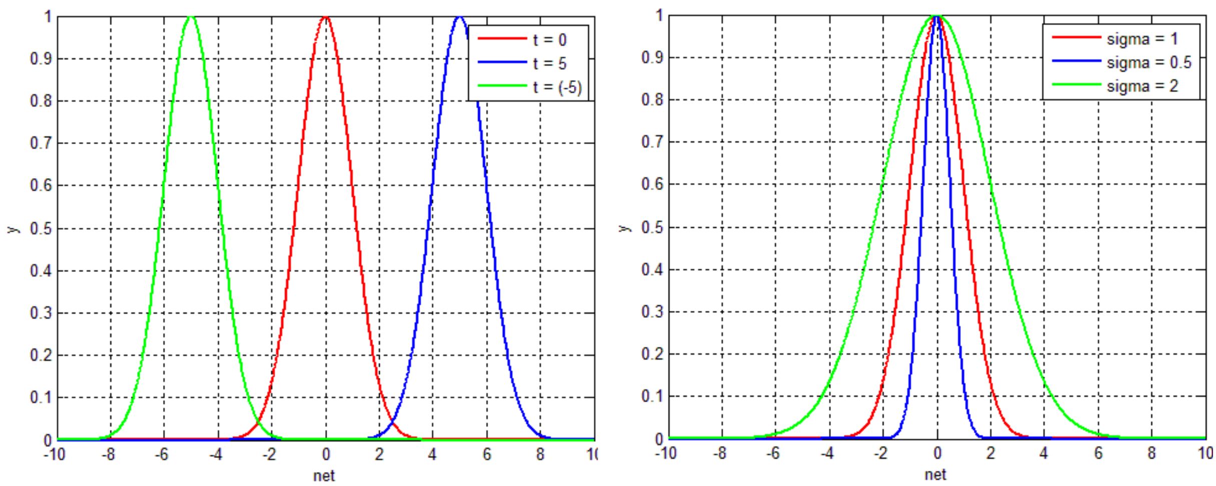
$$y = e^{-\frac{1}{2}\left(\frac{net-t}{\sigma}\right)^2} \tag{2.7}$$




**2.5. Gaussova funkcija**

U jednadžbi (2.7) $net$, kao i ranije, predstavlja sumu otežanih ulaza u neuron, $t$ predstavlja poziciju centra radijalne Gaussove funkcije, a $\sigma$ predstavlja širinu 'zvona'.

Pri korištenju ove funkcije u neuronskoj mreži, uz težine slojeva mijenjat će se i parametri širine i pozicije centra funkcije, pri čemu se pokazuje prednost Gaussove funkcije kao aktivacijske funkcije – neuronska mreža ne treba *Bias* neuron, s obzirom da je pozicija centra $t$ aktivacijske funkcije promjenjiva, kao i širina $\sigma$. Na slici 2.6. prikazana su dva grafa koji pokazuju utjecaj promjene svakog od parametara zasebno za proizvoljne vrijednosti parametara.




**2.6. Grafovi Gaussove funkcije za proizvoljne vrijednosti parametara pozicije centra (lijevo) i širine krivulje (desno)**

Za kasniji izračun gradijenata pogreške potrebno je derivirati funkciju po oba parametra kao što je učinjeno za unipolarnu sigmoidalnu aktivacijsku funkciju.

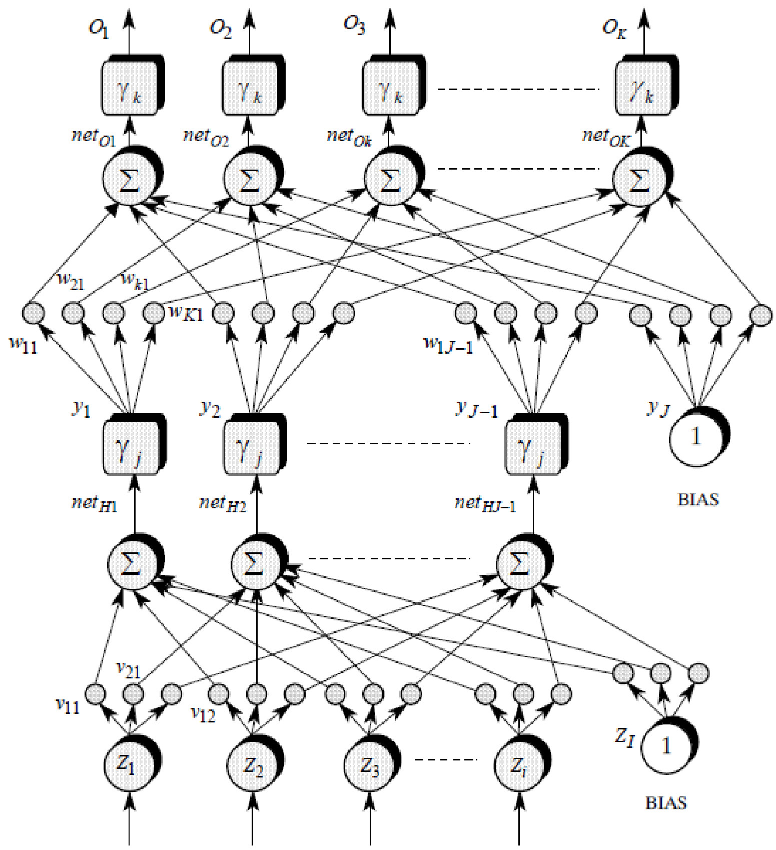
$$y'(net) = \left[\frac{-(net - t)}{\sigma^2}\right] \cdot y \tag{2.8}$$

$$y'(t) = \left(\frac{net - t}{\sigma^2}\right) \cdot y \tag{2.9}$$

$$y'(\sigma) = \left[\frac{(net - t)^2}{\sigma^3}\right] \cdot y \tag{2.10}$$

## 2.4 Model mreže

Kako bi dobili strukturu neuronske mreže, potrebno je neurone organizirati u paralelne slojeve, a slojeve međusobno povezati vezama opterećenim težinskim koeficijentima. Kao što je prije spomenuto, razlikujemo tri tipa slojeva mreže: ulazni sloj, sakriveni sloj i izlazni sloj neurona. Ulazni i izlazni sloj u direktnoj su interakciji s okolinom, dok sakriveni sloj nije, pa mu od tuda i potječe ime. Teoretski radovi nekih istraživača pokazali su kako je jedan sakriveni sloj dovoljan za dobru aproksimaciju bilo koje kontinuirane funkcije [1], uz uvjet da sloj ima dovoljan broj neurona, tako da ću u svom radu mrežu izgraditi sa samo jednim sakrivenim slojem. Nažalost, još nije dokazano koliki je točno broj potreban za određen problem, tako da se broj neurona određuje eksperimentalno, što ću i ja uraditi. Struktura statičke neuronske mreže s jednim skrivenim slojem može se vidjeti na slici 2.7.




**2.7. Struktura statičke neuronske mreže**

### 2.4.1 Unaprijedna faza

Na slici se vidi tok signala kroz neuronsku mrežu i on je slijedeći: Ulazi neurona ulaznog sloja $Z_i$ uz $i = 1,2, \ldots, (I-1)$ su ujedno i ulazi u mrežu, te je ulaz $Z_I$ ulaz *Bias* neurona (uvijek jednak jedinici). Neuroni ulaznog sloja nemaju funkciju sume niti aktivacijsku funkciju, te se stoga ulazni sloj ne smatra pravim slojem i ne broji se u opisivanju slojevitosti neuronske mreže. Umjesto toga neuroni ulaznog sloja služe svojevrsnoj distribuciji ulaznih vrijednosti prema prvom (sakrivenom) sloju. Uz to što su izlazi neurona unutarnjeg sloja spojeni na svaki ulaz sakrivenog sloja, oni su prije toga i pomnoženi sa težinama $v_{ji}$ gdje indeks $j = 1,2, \ldots, (J-1)$ označava broj neurona skrivenog sloja, ne računajući *Bias* neuron smješten u skrivenom sloju (označen indeksom $J$). Funkcija sume neurona skrivenog sloja označena je sa $net_{Hj}$ gdje oznaka $H$ stoji za englesku riječ *hidden* (sakriveni), a $j$ za broj neurona skrivenog sloja, i ona je zbroj svih izlaza neurona ulaznog sloja, ubrajajući *Bias* neuron ulaznog sloja, pomnoženih s pripadajućim težinama skrivenog sloja, šro se može vidjeti iz jednadžbe (2.11).

$$net_{Hj} = \sum_{i=1}^{I} v_{ji} \cdot Z_i \tag{2.11}$$

Slijedeći korak u toku signala je prolazak rezultata funkcije sume $net_{Hj}$ kroz aktivacijsku funkciju $\gamma_j$ pripadajućeg neurona skrivenog sloja (naravno, tu ne ubrajamo *Bias* neuron, čiji je izlaz uvijek jednak jedinici). Izlaz iz neurona skrivenog sloja označavamo sa $y_j$.

$$y_j = \gamma_j\left(net_{Hj}\right) \tag{2.12}$$

U slučaju da koristimo unipolarnu sigmoidalnu aktivacijsku funkciju, njen oblik će biti

$$y_j = \frac{1}{1 + e^{(-C \cdot net_{Hj})}} \tag{2.13}$$

A u slučaju da smo koristili Gaussovu aktivacijsku funkciju, njen oblik će biti

$$y_j = e^{-\frac{1}{2}\left(\frac{net_{Hj} - t_j}{\sigma_j}\right)^2} \tag{2.14}$$

Nakon što smo dobili izlaze iz neurona skrivenog sloja, potrebno ih je spojiti na ulaze posljednjeg, izlaznog sloja, no najprije ih treba pomnožiti sa pripadajućim težinama izlaznog sloja $w_{kj}$. Iznova se spaja izlaz svakog neurona skrivenog sloja na ulaz svakog neurona izlaznog sloja, uz iznimku da izlazni sloj ne posjeduje *Bias* neuron (naravno, prilikom korištenja Gaussove funkcije kao aktivacijske funkcije, niti jedan sloj neće imati *Bias* neuron s obzirom da nije potreban). Funkcija sume $net_{Ok}$ neurona izlaznog sloja je onda

$$net_{Ok} = \sum_{j=1}^{J} w_{kj} \cdot y_j \tag{2.15}$$

Gdje indeks $k = 1,2, \ldots, K$ označava broj neurona izlaznog sloja.

Ako za aktivacijsku funkciju izlaznog sloja odaberemo linearnu funkciju, čime je moguće ostvariti vrijednosti izlaza mreže i veće od jedinice, onda dobivamo aproksimaciju izlaza mreže kako slijedi:

$$O_k = K_p \cdot net_{Ok} \tag{2.16}$$

gdje je $K_p$ nagib linearne funkcije. U nastavku je korišteno $K_p = 1$.

### 2.4.2 Funkcija cilja

U drugoj, povratnoj fazi učenja, na osnovu ostvarenog izlaza mreže i željenog izlaza mreže izračunava se pogreška učenja. Na osnovu pogreške učenja vrši se korekcija vrijednosti težinskih koeficijenata veza među slojevima, kao i parametara aktivacijskih funkcija ($C$ kod unipolarne sigmoidalne funkcije s adaptacijom nagiba, $t$ i $\sigma$ kod Gaussove radijalne funkcije). Čitav postupak se za svaki ulazno-izlazni par podataka zapisnika učenja ponavlja sve dok se ne postigne pogreška manja ili jednaka dozvoljenoj pogrešci (dozvoljeno odstupanje izlaza od željenog izlaza) koju ovisno o zadatku određuje učitelj.

Uobičajena statistička metoda regresijske analize, suma kvadrata pogreške kao mjera odstupanja izlaza mreže od željene vrijednosti izlaza najčešće je korištena funkcija cilja:

$$E = \frac{1}{2}\sum_{n=1}^{N}(d_n - O_n)^2 \tag{2.17}$$

gdje je $N$ broj elemenata u skupu za učenje, odnosno broj podataka učenja iz zapisnika učenja (1/2 je tu kako bi se pojednostavila derivacija funkcije cilja). Prema tome postupak podešavanja težinskih koeficijenata (parametara učenja) je takvo podešavanje težinskih koeficijenata veza koje minimizira funkciju cilja $E$, određenu izrazom (2.17).

### 2.4.3 Ocjena uspješnosti

U radu je, kao uvjet prekida učenja mreže, korišten proizvoljno odabran broj koraka učenja mreže. Ipak, kako bi se procijenila točnost (uspješnost) algoritma učenja u rješavanju postavljenog zadatka, bilo je potrebno definirati mjeru točnosti. Za to je odabran bezdimenzionalni normalizirani korijen srednje kvadratne pogreške (eng. *Normalised Root Mean Square*, ili NRMS).

$$NRMS = \frac{\sqrt{\frac{\sum_{n=1}^{N}(d_n - O_n)^2}{N}}}{\sigma_{d_n}} \tag{2.18}$$

gdje je $\sigma_{d_n}$ standardna devijacija definirana kao:

$$\sigma_{d_n} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(d_n - \bar{d}\right)^2} \tag{2.19}$$

$$\bar{d} = \frac{1}{N}\sum_{n=1}^{N} d_n \tag{2.20}$$

Prednost ove mjere točnosti jest njena bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina, te omogućuje usporedbu izvedenih algoritama učenja s drugim algoritmima, neovisno o korištenoj sklopovskoj ili programskoj podršci.

### 2.4.4 Promjena parametara učenja

Na osnovu odabrane funkcije cilja vrši se promjena koeficijenata težina i parametara aktivacijskih funkcija primjenom nekog od algoritama nelinearnog optimiranja. Izrazom (2.21) dana je poznata forma promjene parametara učenja $\vartheta$ :

$$\vartheta(n+1) = \vartheta(n) + \Delta\vartheta(n) \tag{2.21}$$

gdje je $n$ trenutni korak učenja, $\Delta\vartheta(n)$ veličina promjene parametara učenja , a $\vartheta(n+1)$ je nova vrijednost parametra učenja. Pogrešku $E(\vartheta)$ moguće je u okolišu točke $\vartheta$ aproksimirati s prva dva člana Taylorovog reda:

$$E(\vartheta + \Delta\vartheta) \approx E(\vartheta) + \Delta E(\vartheta) \tag{2.22}$$

$$\Delta E(\vartheta) = \Delta\vartheta^T \nabla E(\vartheta) \tag{2.23}$$

$$\nabla E(\vartheta) = \frac{\delta E(\vartheta)}{\delta\vartheta} \tag{2.24}$$

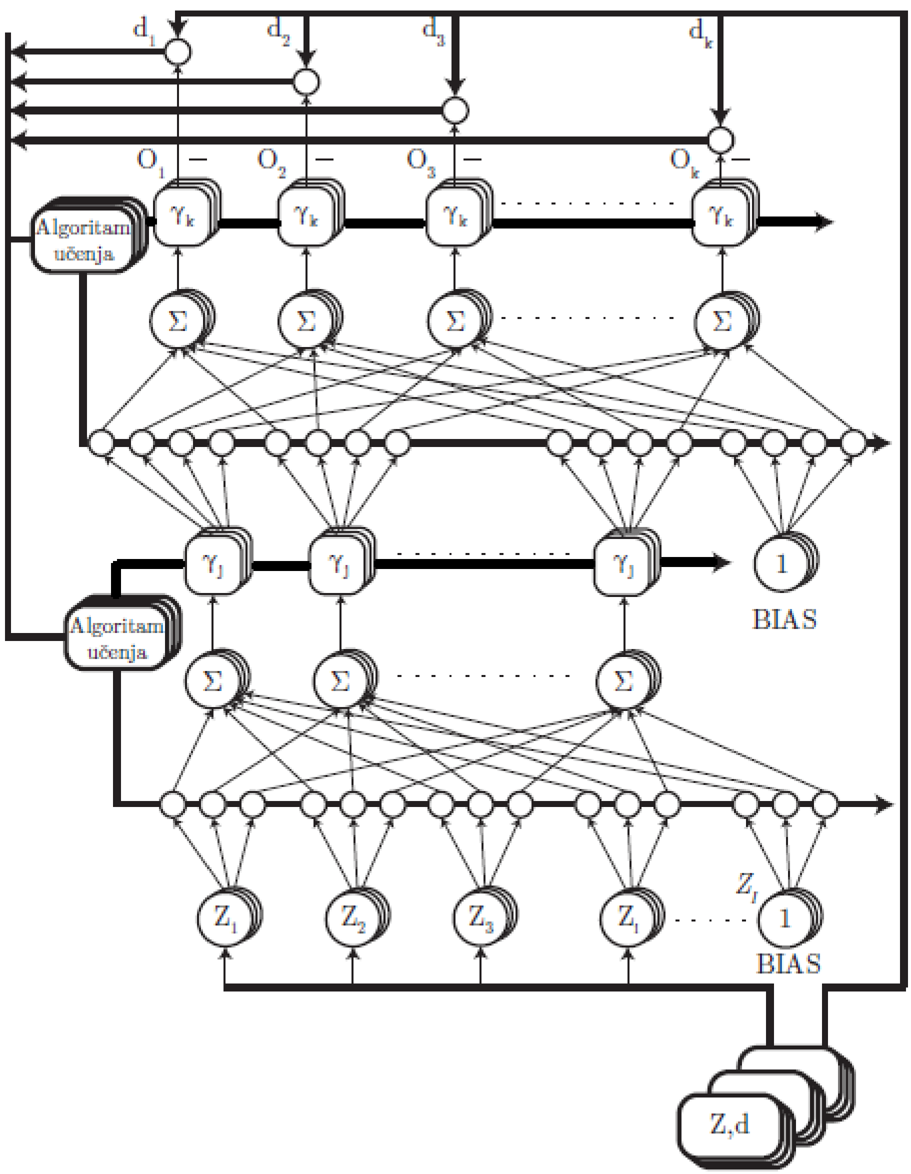
Izraz (2.24) naziva se gradijentom pogreške. Da bi se pogreška smanjivala najvećim mogućim iznosom, treba odrediti $\Delta\vartheta$ za koji promjena pogreške učenja $\Delta E(\vartheta)$ poprima najveći negativni iznos, a to se ostvaruje uz uvjet:

$$\Delta\vartheta = -\eta \nabla E(\vartheta) \tag{2.25}$$

gdje je $\eta$ mjera te promjena, koja se još naziva i koeficijentom brzine učenja. Koeficijen određuje učitelj, te mu se vrijednost najčešće odabire između 0.001 i 10. Izraz (2.25) osigurava promjenu parametara učenja u smjeru najstrmijeg pada ukupne pogreške. Uvrštavanjem izraza (2.25) u (2.21) dobiva se:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)) \tag{2.26}$$

Algoritam dan izrazom (2.23) poznat je u literaturi o nelinearnom optimiranju pod nazivom algoritam najstrmijeg pada (eng. *Steepest Descent*), dok je prema literaturi o neuronskim mrežama poznat pod nazivom algoritam povratnog prostiranja pogreške (eng. Error Backpropagation Algorithm). Na slici 2.8. možemo vidjeti shematski prikaz neuronske mreže te način mijenjanja parametara učenja EBP algoritmom.




**2.8. Shema promjene parametara učenja algoritmom učenja**

Važno je napomenuti da će unaprijedna faza učenja biti identična za svaku mrežu, neovisno o aktivacijskoj funkciji koju koristimo, no povratna je faza različita za svaku, s obzirom da su i derivacije potrebne za izračun gradijenta različite. Također, treba spomenuti kako se promjena parametara učenja može u osnovi odvijati na dva načina – učenjem po uzorku, gdje se parametri mijenjaju nakon prolaska svakog ulazno-izlaznog para skupa za učenje, na temelju pogreške dobivene prolaskom samo jednog ulazno-izlaznog para kroz mrežu; te učenjem po skupu, gdje se parametri mijenjaju tek nakon što cijeli ulazno-izlazni skup za učenje prođe kroz mrežu, na temelju srednje pogreške svih datoteka ulazno-izlaznog skupa za učenje. Algoritmi u ovom radu napravljeni su po principu učenja po uzorku.

Dakle, ako smo odabrali **unipolarnu sigmoidalnu aktivacijsku funkciju**, promjena parametara učenja se odvija po slijedećim izrazima.

Težine izlaznog sloja:

$$\frac{\delta E}{\delta w_{kj}} = \frac{\delta E}{\delta O_k} \frac{\delta O_k}{\delta net_{Ok}} \frac{\delta net_{Ok}}{\delta w_{kj}} = -(d_k - O_k) \cdot y_j \tag{2.27}$$

$$w_{kj}(n+1) = w_{kj}(n) - \eta \frac{\delta E}{\delta w_{kj}} \tag{2.28}$$

Težine skrivenog sloja:

$$\frac{\delta E}{\delta v_{ji}} = \frac{\delta E}{\delta y_j} \frac{\delta y_j}{\delta net_{Hj}} \frac{\delta net_{Hj}}{\delta v_{ji}} = C \cdot y_j(1 - y_j) \left( -\sum_{k=1}^{K} (d_k - O_k) w_{kj} \right) \cdot Z_i \tag{2.29}$$

$$v_{ji}(n+1) = v_{ji}(n) - \eta \frac{\delta E}{\delta v_{ji}} \tag{2.30}$$

Koeficijent nagiba sigmoidalne funkcije (u slučaju aktivacijske funkcije u kojem se i on mijenja uz ostale parametre):

$$\frac{\delta E}{\delta C_j} = \frac{\delta E}{\delta y_j} \frac{\delta y_j}{\delta C_j} = net_{Hj} \cdot y_j(1 - y_j) \left( -\sum_{k=1}^{K} (d_k - O_k) w_{kj} \right) \tag{2.31}$$

$$C_j(n+1) = C_j(n) - \eta \frac{\delta E}{\delta C_j} \tag{2.32}$$

U slučaju neuronske mreže sa **Gaussovom radijalnom aktivacijskom funkcijom** parametri učenja se mijenjaju po slijedećim izrazima:

Težine izlaznog sloja:

$$\frac{\delta E}{\delta w_{kj}} = \frac{\delta E}{\delta O_k} \frac{\delta O_k}{\delta net_{Ok}} \frac{\delta net_{Ok}}{\delta w_{kj}} = -(d_k - O_k) \cdot y_j \tag{2.33}$$

$$w_{kj}(n+1) = w_{kj}(n) - \eta \frac{\delta E}{\delta w_{kj}} \tag{2.34}$$

Težine skrivenog sloja:

$$\frac{\delta E}{\delta v_{ji}} = \frac{\delta E}{\delta y_j} \frac{\delta y_j}{\delta net_{Hj}} \frac{\delta net_{Hj}}{\delta v_{ji}} = \left( -\sum_{k=1}^{K} (d_k - O_k) w_{kj} \right) \left[ \frac{-(net - t)}{\sigma^2} \right] y_j Z_i \tag{2.35}$$

$$v_{ji}(n+1) = v_{ji}(n) - \eta \frac{\delta E}{\delta v_{ji}} \tag{2.36}$$

Centar Gaussove funkcije:

$$\frac{\delta E}{\delta t_j} = \frac{\delta E}{\delta y_j}\frac{\delta y_j}{\delta t_j} = \left(-\sum_{k=1}^{K}(d_k - O_k)w_{kj}\right)\left(\frac{net - t}{\sigma^2}\right)y_j \tag{2.37}$$

$$t_j(n+1) = t_j(n) - \eta\frac{\delta E}{\delta t_j} \tag{2.38}$$

Širina Gaussove funkcije:

$$\frac{\delta E}{\delta \sigma_j} = \frac{\delta E}{\delta y_j}\frac{\delta y_j}{\delta \sigma_j} = \left(-\sum_{k=1}^{K}(d_k - O_k)w_{kj}\right)\left[\frac{(net - t)^2}{\sigma^3}\right]y_j \tag{2.39}$$

$$\sigma_j(n+1) = \sigma_j(n) - \eta\frac{\delta E}{\delta \sigma_j} \tag{2.40}$$

Učenje neuronske mreže po uzorku će se odvijati na slijedeći način – jedan po jedan ulazno-izlazni par skupa za učenje prolazi kroz mrežu unaprijedno. Za svaki par se nakon toga u povratnoj fazi računa greška izlaza i gradijenti pogreške parametara učenja, te mijenjaju njihove vrijednosti prema izrazima (2.27) do (2.32) za slučaj da se koristimo unipolarnom sigmoidalnom aktivacijskom funkcijom, ili izrazima (2.33) do (2.40) za slučaj da se koristimo Gaussovom radijalnom aktivacijskom funkcijom. Dakle, u jednom koraku učenja, svi parametri učenja će se promijeniti onoliko puta koliko imamo ulazno-izlaznih parova.

Promjena parametara učenja neuronske mreže odvijati će se iterativno dok stvarni izlaz mreže ne zadovolji jedan od uvjeta koje je zadao učitelj – proizvoljan broj koraka, ili greška manja ili jednaka od neke zadane vrijednosti. Prilikom rješavanja problema za testiranje, mrežama je zadan proizvoljan, no jednak broj koraka, te su se one onda uspoređivale prema izračunatom NRMS-u prema izrazu (2.18).

# 3. Problemi za testiranje

Nakon što se izgradi model neuronske mreže i algoritam učenja, te nakon što se ustanovi funkcija cilja i ocjena uspješnosti, neuronska se mreža obično testira na nekom od standardnih problema. Obično se radi o problemima linearno neseparabilnih uzoraka, predikciji, aproksimaciji ili interpolaciji nekakvih, vrlo često nelinearnih, funkcija.

U ovom radu izgrađena statička unaprijedna umjetna neuronska mreža testirati će se na tri problema za tri različite aktivacijske funkcije – unipolarnom sigmoidalnom funkcijom sa i bez adaptacije nagiba, opisane izrazom (2.13), te Gaussovom radijalnom funkcijom sa adaptibilnim parametrima (pozicija centra i širina krivulje) opisanom izrazom (2.14).

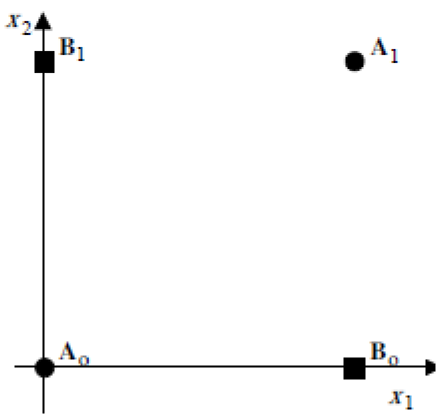
Prvi problem je XOR (ekskluzivni ILI) koji spada u kategoriju problema linearno neseparabilnih uzoraka. Drugi problem je predikcija ponašanja nelinearnog kaotičnog sustava, koji će biti predstavljen Glass-Mackey-evom jednadžbom. Treći je problem učenje ponašanja jednostavnog linearnog dinamičkog člana prvog reda, poznatog pod nazivom $P_1$ ili $P_{T1}$ dinamički član.

## 3.1 XOR problem

Ekskluzivni ILI (eng. XOR), poznati je logički problem Booleove algebre, često primjenjivan za ispitivanje svojstava različitih modela umjetnih neuronskih mreža. Kod ovog jednostavnog problema, ulaz je određen s dvije binarne varijable koje mogu poprimiti vrijednost nula ili jedan. Izlaz se sastoji od jedne binarne varijable, koja treba poprimiti vrijednost jedan ukoliko je samo jedna od binarnih ulaznih varijabli jednaka jedan, odnosno nula ukoliko oba dvije ulazne varijable imaju istu vrijednost (nula ili jedan).

**Tablica 3.1 XOR problem - tablica istine.**

| Točka | $x_1$ | $x_2$ | Željeni izlaz |
|---|---|---|---|
| $A_o$ | 0 | 0 | 0 |
| $B_o$ | 0 | 1 | 1 |
| $B_1$ | 1 | 0 | 1 |
| $A_1$ | 1 | 1 | 0 |




**3.1 Linearno neseparabilni problem**

Uzorci XOR problema definirani tablicom 3.1., grafički su prikazani slikom 3.1. Skupine A i B linearno su neseparabilne (nije moguće povući pravac kojim bi se razdijelili uzorci skupina A i B), te je dobro znano da ovaj problem nije moguće riješiti primjenom jednoslojnih perceptronskih mreža. Isti problem rješiv je uvođenjem nelinearnog skrivenog sloja, odnosno primjenom višeslojnih perceptronskih mreža.

## 3.2 Nelinearni kaotični sustav (Glass-Mackeyeva jednadžba)

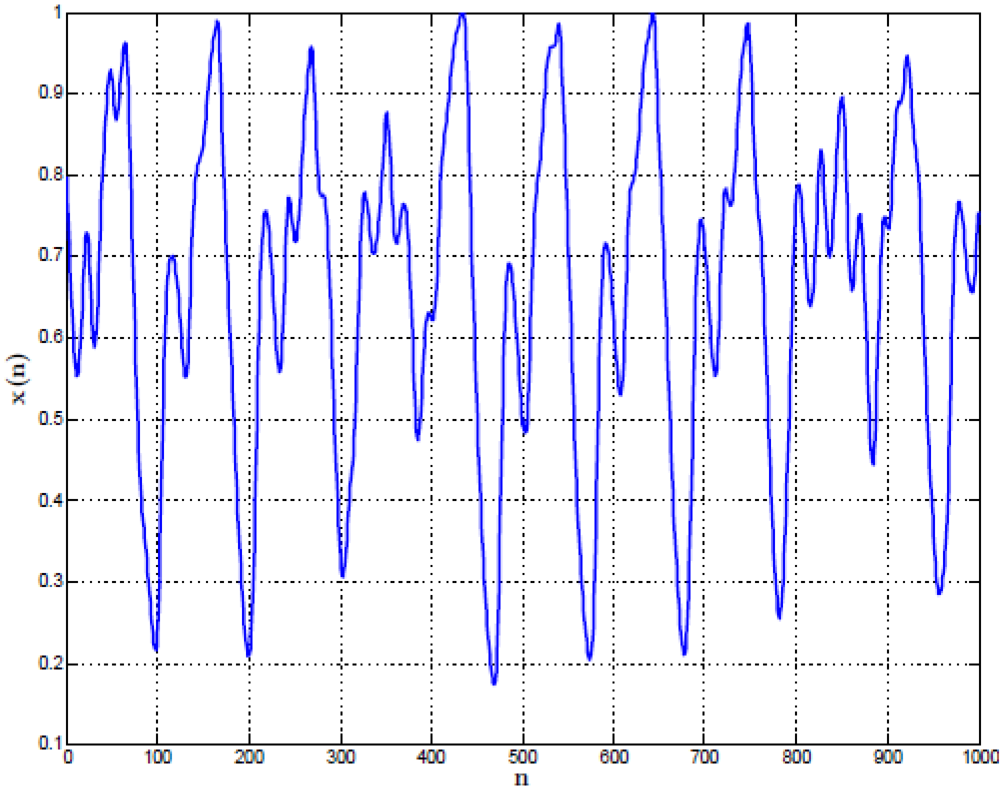
Tradicionalne statističke metode obrade signala daju slabe rezultate u predikciji dinamičkih sustava, a poseban problem su sustavi s nelinearnostima, vremenski promjenjivim parametrima i kaotični sustavi. Poznato je da kaos ima realnu fizikalnu osnovu, a u izvornom se obliku javlja u kemijskim reaktorima i plazmi, zatim kod turbulencije fluida, lasera i slično. Promjenom parametara kaotični sustavi pokazuju široku lepezu najrazličitijih nelinearnih ponašanja, te stoga predstavljaju zgodan problem za ispitivanje raznih tehnika obrade signala, među kojima posebno mjesto zauzimaju neuronske mreže.

Lapedes i Farber [1] predlažu Glass-Mackeyev nelinearni kaotični sustav za testiranje algoritama učenja neuronskih mreža, jer ima jednostavnu definiciju, a njegovo ponašanje teško se predviđa (vremenske serije signala su kaotične). Glass-Mackeyeva jednadžba dana izrazom (3.1) nelinearna je diferencijalna jednadžba sa kašnjenjem za interval $\tau$, koja je određena početnom (inicijalnom) funkcijom definiranom kako slijedi:

$$\dot{x} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \tag{3.1}$$

gdje su $a$ i $b$ parametri sustava, a $t$ oznaka vremena. Prikaz vremenske serije od 1000 točaka za $\tau = 30$ dan slikom 3.2., dobiven je simulacijom izraza (3.2) u diskretnoj formi, kako bi bilo moguće raditi s tim izrazom na računalu. Za period uzrokovanja odabrana je 1 $s$, čime je dobiveno $t = n$, $n = 0,1,2,\ldots$ Početna vrijednost sustava postavljena je na $x_0 = 1.2$, dok je za vremensko kašnjenje odabrana vrijednost $\tau = 30$. Parametri $a$ i $b$ postavljeni su na vrijednost 0.2, odnosno 0.1. Diskretizirana jednadžba prikazana je izrazom (3.2).

$$x(n) = \frac{1}{1+b}\left[x(n-1) + \frac{ax(n-\tau)}{1+x^{10}(n-\tau)}\right] \tag{3.2}$$




**3.2 Graf diskretiziranog izraza za Glass-Mackey sustav**

Predviđanje ponašanja nelinearnog kaotičnog sustava svodi se na pronalazak funkcije preslikavanja $f(\bullet)$ koja će temeljem prethodnih vrijednosti signala omogućiti predviđanje vrijednosti signala nekoliko koraka unaprijed.

$$x(n+P) = f(\, x(n), x(n-\Delta), x(n-2\Delta), \dots, x(n-m\Delta)) \tag{3.3}$$

gdje je $P$ broj točaka signala koji se želi predvidjeti unaprijed, $\Delta$ predstavlja kašnjenje signala, a $m$ je cjelobrojna konstanta. U ovom radu odabrano je $P = \Delta = 6$, što znači da se predviđa šest točaka unaprijed na osnovu trenutne vrijednosti signala i vrijednosti signala u prethodnim točkama, višekratnicima broja šest. Također, odabrano je da se predviđanje ostvaruje na temelju četiri prethodne vrijednosti signala, $m = 4$, pa sada izraz (3.3) možemo pisati kao:

$$x(n+6) = f(\, x(n), x(n-6), x(n-12), x(n-18), x(n-24)) \tag{3.4}$$

Funkcija (3.4) ostvarena je neuronskom mrežom. Budući da je broj neurona ulaznog i sakrivenog sloja neuronske mreže ovisan o ulazno-izlaznom uzorku, tako u slučaju predviđanja ponašanja nelinearnog kaotičnog sustava mreža ima pet ulaznih neurona (pet je ulaza - jedna točka iz trenutnog koraka, i četiri točke iz prethodnih koraka) i jedan izlazni neuron (predviđa se jedna točka).

Zbog korištenja unipolarne sigmoidalne funkcije kao aktivacijske funkcije neurona sakrivenog sloja, u ulazni i sakriveni sloj neuronske mreže u kojoj se takva aktivacijska funkcija koristi potrebno je dodati po jedan *Bias* neuron. U mreži u kojoj koristima Gaussovu aktivacijsku funkciju takvo što se ne mora raditi.

Učenje i testiranje mreže ostvareno je podjelom vremenske serije (slika 3.2.) na dva jednaka dijela - prvih 500 točaka korišteno je se za učenje, a potom je rad naučene mreže testiran na tri niza od 500 točaka koji su se razlikovali po početnim stanjima.

# 4. Rezultati

Nakon što je postavljen matematički model neuronske mreže, u programskom paketu *MATLAB* su prema modelu izrađene tri verzije mreže za tri aktivacijske funkcije kojima se bavi ovaj rad – unipolarna sigmoidalna funkcija sa i bez adaptacije nagiba, te Gaussova funkcija sa adaptibilnim parametrima. Sva testiranja obavljena su uz pomoć tih mreža. Broj ulaza i izlaza, odnosno broj ulaznih i izlaznih neurona, ovisio je o problemu.

## 4.1. Rezultati XOR problema

U slučaju XOR problema broj neurona ulaznog sloja morao je biti dva, s obzirom da su ulazi u sustav binarni logički parovi. U mrežama koje su koristile unipolarnu sigmoidalnu funkciju dodan je i *Bias* neuron sa izlazom jednakim jedinici. Broj neurona izlaznog sloja je jednak jedinici, s obzirom da XOR mora davati odgovor u obliku jedne binarne znamenke. Broj neurona skrivenog sloja sam odlučio zadati kao dva, uz dodatak *Bias* neurona za mreže sa unipolarnom sigmoidalnom aktivacijskom funkcijom. Svakoj mreži zadan je identičan broj koraka učenja $i = 1000$, te se mreže ocjenjivalo po vrijednosti NRMS-a koje su uspjele ostvariti. Početne vrijednosti parametara učenja zadane su $randi(\quad)$ funkcijom programskog paketa *MATLAB,* te su generirane veličine bile u intervalu između 0 i 1 i jednake za sve mreže ondje gdje su te mreže dijelile iste parametre učenja (težine slojeva). Kod Gaussove funkcije početne širine funkcije bile su $\sigma_j = 0.5$ za sve neurone, a pozicije centra su postavljene u nuli. Kod unipolarne sigmoidalne funkcije početni koeficijenti nagiba sigmoidalne funkcije postavljeni su u jedinicu, kod sigmoidalne funkcije bez adaptacije nagiba ostali su jednaki jedan. Koeficijent brzine učenja $\eta$ za izmjenu svih parametara je bio $\eta = 0.1$.

Tablica 4.1 uspoređuje tražene izlaze za zadane ulaze sa stvarnim izlazima svake od mreža.

**4.1 Usporedba stvarnih i traženih izlaza za različite aktivacijske funkcije**

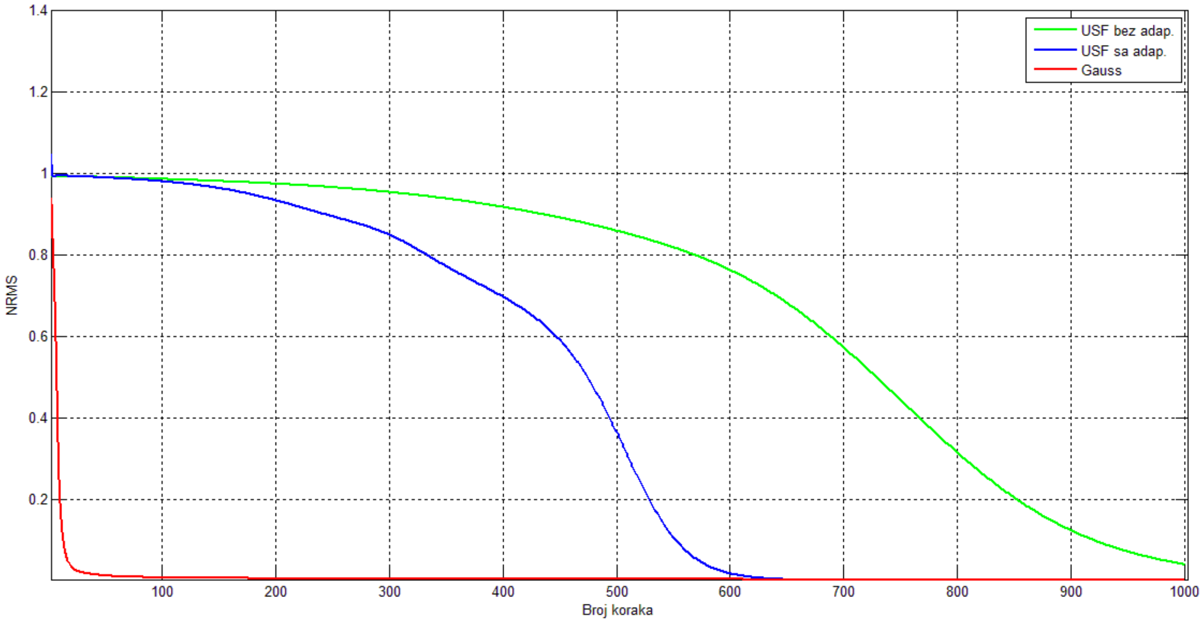
| Ulaz 1 | Ulaz 2 | Traženi izlaz | USF bez adap. | USF sa adap. | Gauss |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0.01644 | 0.00000 | 0.00035 |
| 0 | 1 | 1 | 0.98184 | 0.99999 | 0.99891 |
| 1 | 0 | 1 | 0.97526 | 0.99999 | 0.99876 |
| 1 | 1 | 0 | 0.01839 | 0.00000 | 0.00031 |

Tablica 4.2. uspoređuje izračunate NRMS-e za sve tri mreže s različitim aktivacijskim funkcijama u posljednjem koraku učenja.

**4.2 Usporedba NRMS-a za mreže s tri aktivacijske funkcije skrivenog sloja za posljednji korak**

| | NRMS | korak |
|---|---|---|
| **USF bez adaptacije nagiba** | 0.039370 | 1000 |
| **USF sa adaptacijom nagiba** | 1.53274e-09 | 1000 |
| **Gaussova radijalna** | 0.001709 | 1000 |

Već se iz te tablice može vidjeti kvaliteta svake aktivacijske funkcije – unipolarna sigmoidalna funkcija s adaptacijom nagiba je najbolja, s obzirom da ostvaruje najmanju grešku; nakon nje slijede Gaussova funkcija, te unipolarna sigmoidalna funkcija bez adaptacije nagiba. Na slici 4.1. prikazan je graf konvergencije ka točnom rješenju tokom učenja mreže. To je postignuto tako da se napravi graf s NRMS-om na ordinati, i brojem koraka na apscisi, što nam pokazuje kako se greška postepeno smanjuje.




**4.1. Graf NRMS za sve tri aktivacijske funkcije**

Na slici 4.1. se može vidjeti da je situacija zapravo puno drugačija od one koja se mogla pretpostaviti iz Tablice 4.2. Iako je neuronska mreža s Gaussovom aktivacijskom funkcijom u konačnici dala netočnije rezultate od one s unipolarnom sigmoidalnom funkcijom s adaptacijom nagiba, iz grafa na slici se može vidjeti kako je zato daleko brže konvergirala rješenju, već negdje oko 20-og koraka učenja, dok je druga mreža to postigla tek negdje oko 600-tog koraka.

Iz toga se može reći kako je neuronska mreža koja koristi Gaussovu aktivacijsku funkciju s adaptibilnim parametrima zapravo puno bolja u rješavanju XOR problema od ostale dvije mreže. Uzevši u obzir i činjenicu da traženi izlazi XOR problema ne mogu posjedovati niti jednu drugu vrijednost osim 0 i 1, te da preciznost stvarnih izlaza mreže zapravo nije niti potrebna dalje od druge decimale, zaključuje se kako je neuronska mreža koja koristi Gaussovu funkciju s adaptibilnim parametrima kao aktivacijsku u skrivenom sloju doista najbolja u rješavanju XOR problema u zadanim uvjetima, te sa zadanom metodom ocjene uspješnosti. Nakon nje slijedi mreža s unipolarnom sigmoidalnom funkcijom sa promjenjivim koeficijentom nagiba koja, kao što je prije spomenuto, konvergira oko 600-tog koraka, te kao posljednja i najlošija dolazi mreža s unipolarnom sigmoidalnom aktivacijskom funkcijom bez mogućnosti adaptacije nagiba, koja niti u 1000-itom koraku nije uspjela postići točnost kao i ostale dvije, iako se vidi kako konvergencija nije bila daleko (možda negdje oko 1100-tog koraka).

## 4.2 Rezultati predikcije Glass-Mackey kaotičnog sustava

U ovom slučaju broj ulaznih neurona mora biti jednak pet, budući da za dobru predikciju šest koraka unaprijed, uzimamo četiri prethodne vrijednosti i onu trenutnu, kao što je opisano u poglavlju 3.2 i izrazu (3.4). Izlaz jednadžbe je samo jedan, tako da će mreža imati samo jedan izlazni neuron. Odabir broja neurona je, kao što je objašnjeno u poglavlju 2.4, proizvoljan do određene mjere. Rečeno je da neuronska mreža može zadovoljavajuće aproksimirati bilo koju kontinuiranu funkciju sa samo jednim skrivenim slojem, no eksperimentalno se mora odrediti broj neurona koji sačinjavaju taj sloj. Pritom se treba izbjeći korištenje prevelikog broja neurona, kako bi se izbjegla pretreniranost, pojam koji označava mrežu koja vrlo brzo konvergira, ali ima veoma lošu sposobnost generalizacije. U ovom radu problem se pokušao riješiti sa pet neurona skrivenog sloja. Naravno, kao i ranije, kad mreža koristi unipolarne sigmoidalne aktivacijske funkcije ulaznom i skrivenom sloju je potrebno dodati *Bias* neuron.

Početne težine su se opet zadale slučajno, pomoću MATLAB-ove funkcije $randi(\quad)$, u intervalu između nula i jedan. Mreža koja je koristila unipolarnu sigmoidalnu aktivacijsku funkciju sa adaptacijom nagiba za početni iznos koeficijenta nagiba funkcije imala je jedinicu, a ona koja nije u mogućnosti adaptirati nagib, imala je fiksnu vrijednost jedan. Mreža koja koristi Gaussovu aktivacijsku funkciju u skrivenom sloju imala je početne vrijednosti širine krivulje $\sigma = 0.5$, a početne vrijednosti centara funkcije bile su namještene u nuli.

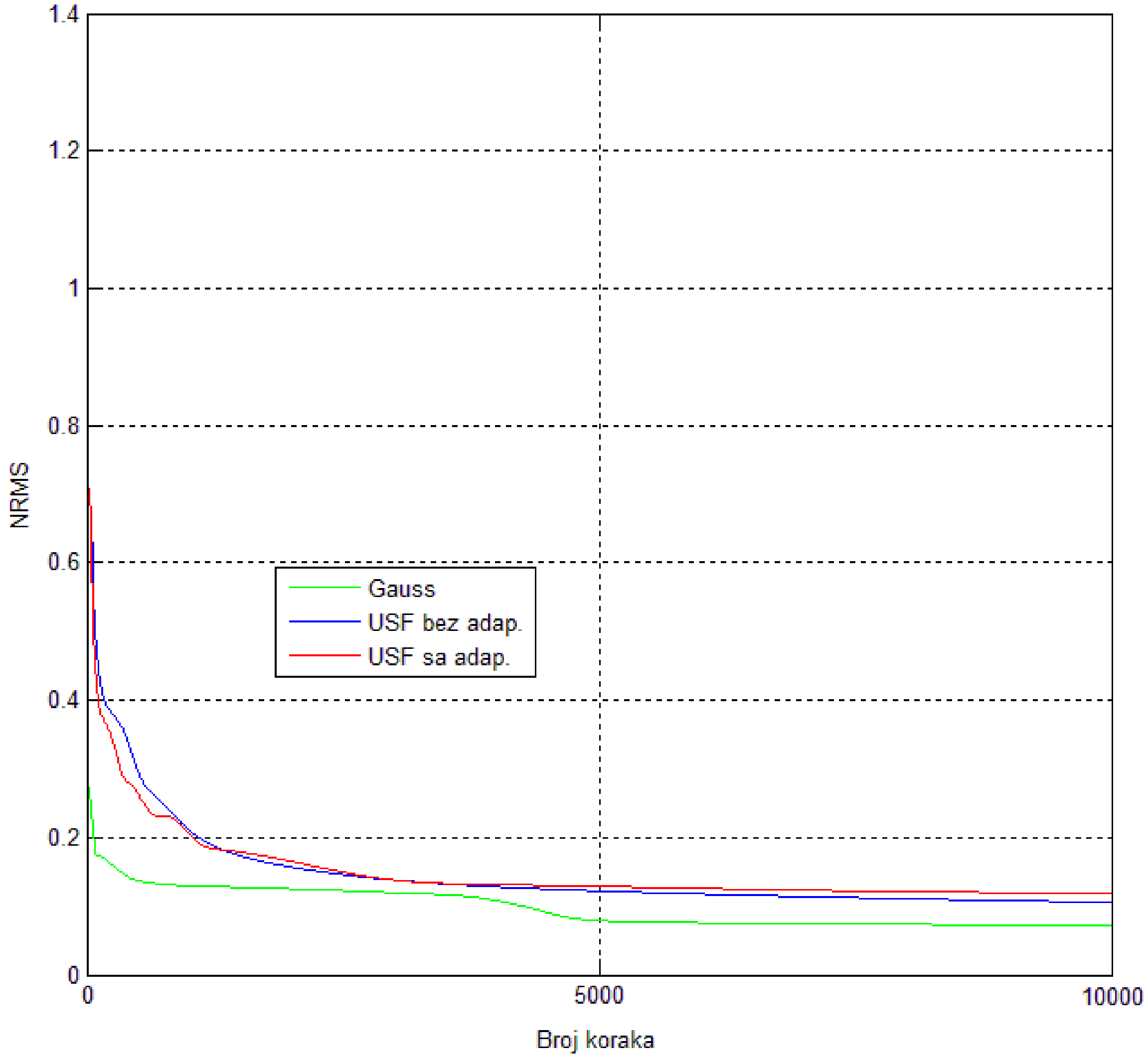
Svaka mreža je prošla 10000 koraka učenja sa koeficijentom brzine učenja za promjenu svih parametara $\eta = 0.009$.

Učenje mreže se izvodi s prvih 500 točaka koje pripadaju vremenskoj seriji prikazanoj slikom 3.2., čija je diskretna forma izražena u (3.2) uz početne uvjete navedene u poglavlju 3.2. Zatim je mreža testirana sa po 500 točaka na tri različita testa koje nije učila. Ocjena uspješnosti se izvodi pomoću NRMS-a koji se računa prema izrazu (2.18).

Tablica 4.3. uspoređuje izračunate NRMS-e za sve tri mreže s različitim aktivacijskim funkcijama za točke na kojima je učila.

**4.3 Usporedba NRMS-a za točke koje je mreža učila za tri aktivacijske funkcije skrivenog sloja za posljednji korak**

| | **NRMS** | **korak** |
|---|---|---|
| **USF bez adaptacije nagiba** | 0.105398 | 10000 |
| **USF sa adaptacijom nagiba** | 0.118020 | 10000 |
| **Gaussova radijalna** | 0.071799 | 10000 |

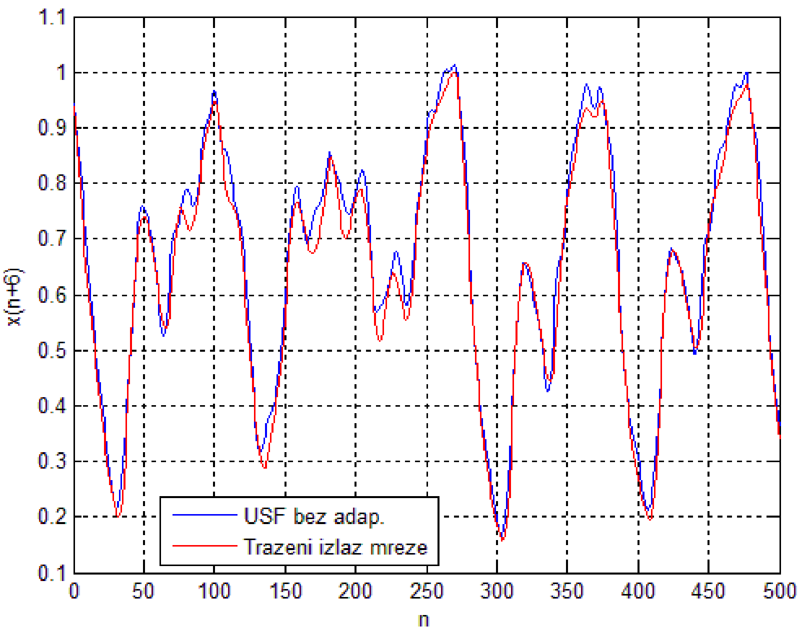



**4.2. Graf promjene NRMS-a za učeni skup podataka**

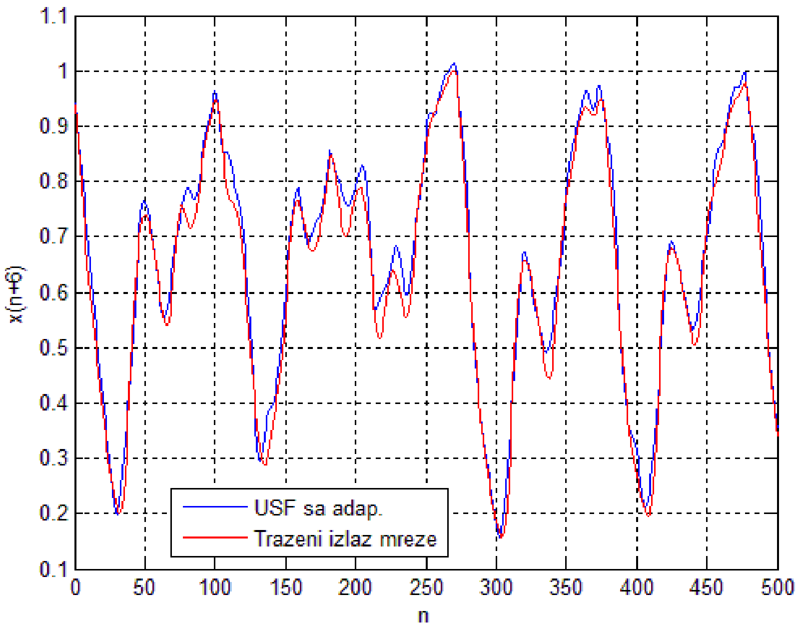
Iz grafa se vidi kako su konvergencije NRMS-a za obje unipolarne sigmoidalne aktivacijske funkcije vrlo slične, s tim da je mreža koja može adaptirati nagib tek ponešto brža. Mreža koja je implementirala Gaussovu aktivacijsku funkciju s druge strane, pokazuje karakteristični oblik koji je vidljiv i u XOR problemu – vrlo brza konvergencija, nakon čega slijedi dugačak dio gdje se greška gotovo i ne smanjuje, što sugerira da nakon ostvarene određene točnosti stvarnog izlaza, mreža nailazi na velike poteškoće u poboljšanju performansa, moguće zato što je zapela u lokalnom minimumu. Takav se problem može djelomice riješiti uvođenjem dodatnih članova u izraze za promjene parametara učenja mreže. Pri takvoj modifikaciji algoritma dodatni član se naziva momentum ili zamah, a efekt mu je ubrzanje algoritma (i do deset puta[1]), no on nije implementiran u ovom radu.

Spomenuto je kako su mreže nakon učenja testirane sa tri različita skupa podataka koji pripadaju vrijednostima Glass-Mackey sustava za različite početne uvjete. Na slikama 4.3. do 4.11. mogu se vidjeti traženi izlazi mreže uz stvarne izlaze svake od mreža. Na svakom od tih testova vidljiva je slična stvar kao i iz slike 4.2. – neuronska mreža koja koristi Gaussovu aktivacijsku funkciju skrivenog sloja radi najbolju predikciju za sva tri testa, dok su mreže s unipolarnim sigmoidalnim aktivacijskim funkcijama nešto lošije i otprilike izjednačene, osim na dijelovima gdje se vrijednosti mijenjaju dosta brzo; tu se vidi prednost neuronske mreže koja može mijenjati nagib sigmoidalne aktivacijske funkcije. Razlog leži u tome što neuronska mreža s Gaussovom aktivacijskom funkcijom posjeduje više adaptibilnih parametara od mreža s drugim funkcijama, pa ima veću sposobnost generalizacije, no zato za njen postupak učenja treba puno više vremena.

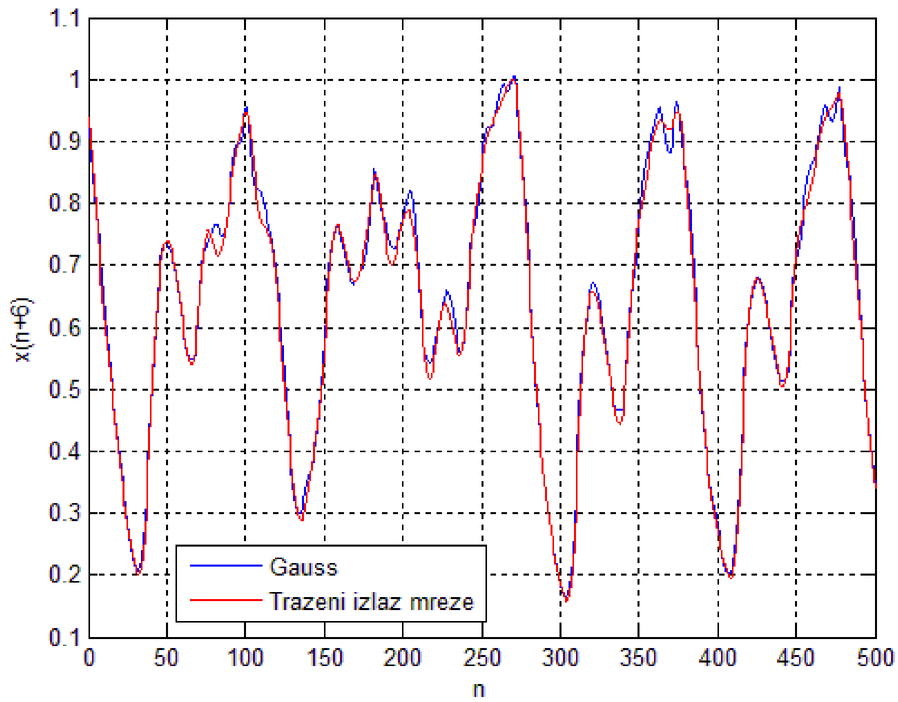
### 4.2.1 Test 1




**4.3. Odaziv mreže s USF aktivacijskom funkcijom bez adaptacije nagiba za test 1**

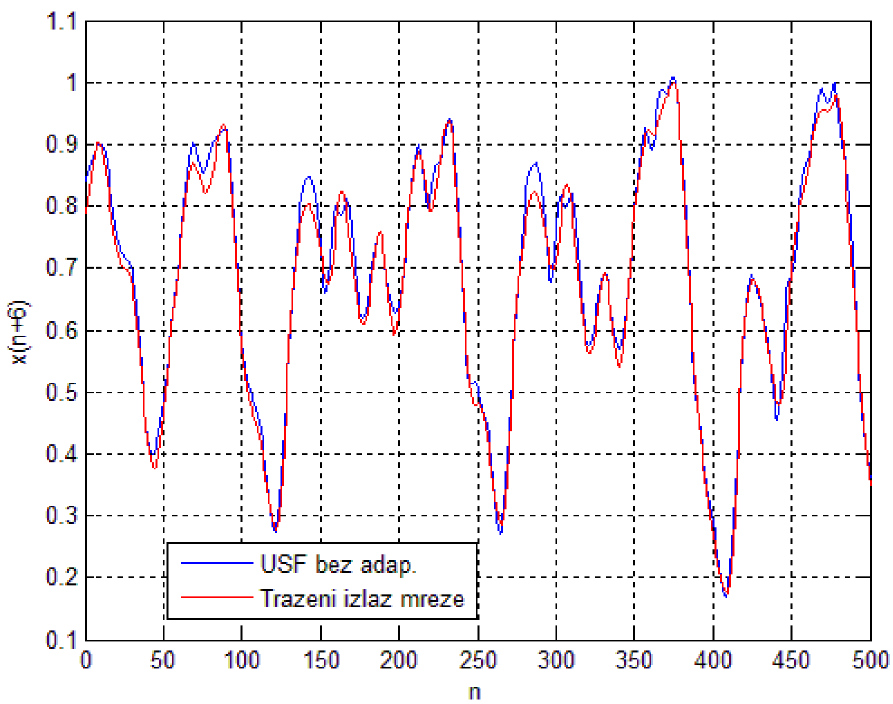



**4.4. Odaziv mreže s USF aktivacijskom funkcijom sa adaptacijom nagiba za test 1**

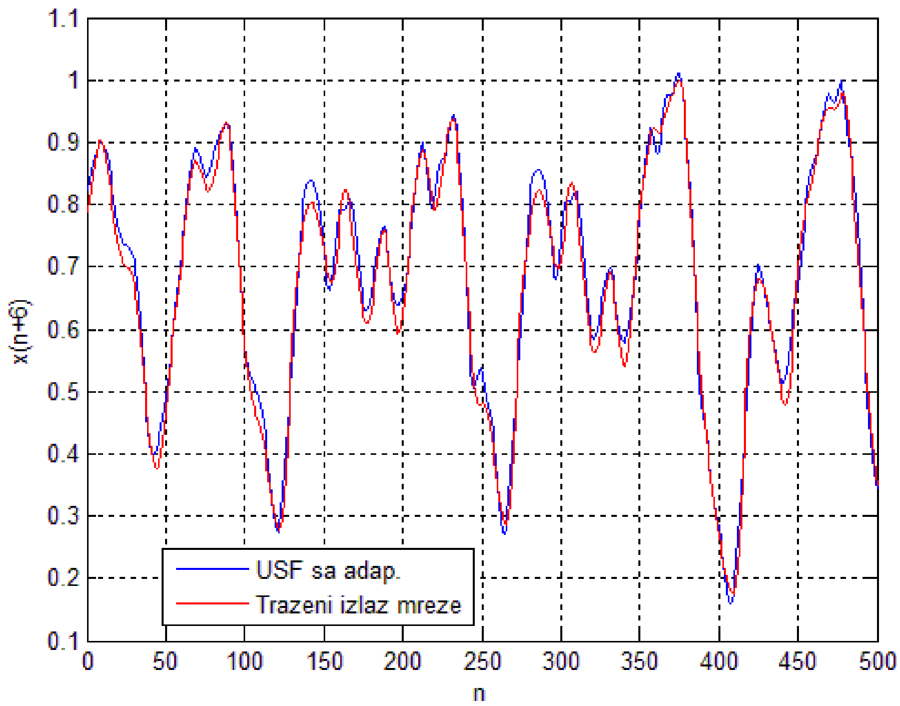



**4.5. Odaziv mreže s Gaussovom aktivacijskom funkcijom za test 1**

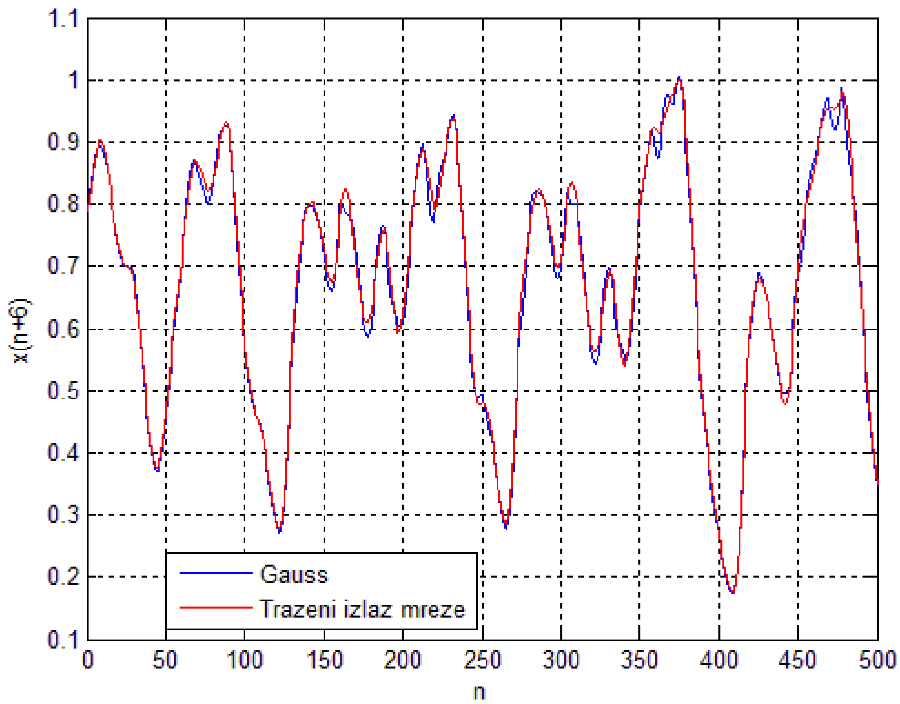
### 4.2.2 Test 2




**4.6. Odaziv mreže s USF aktivacijskom funkcijom bez adaptacije nagiba za test 2**

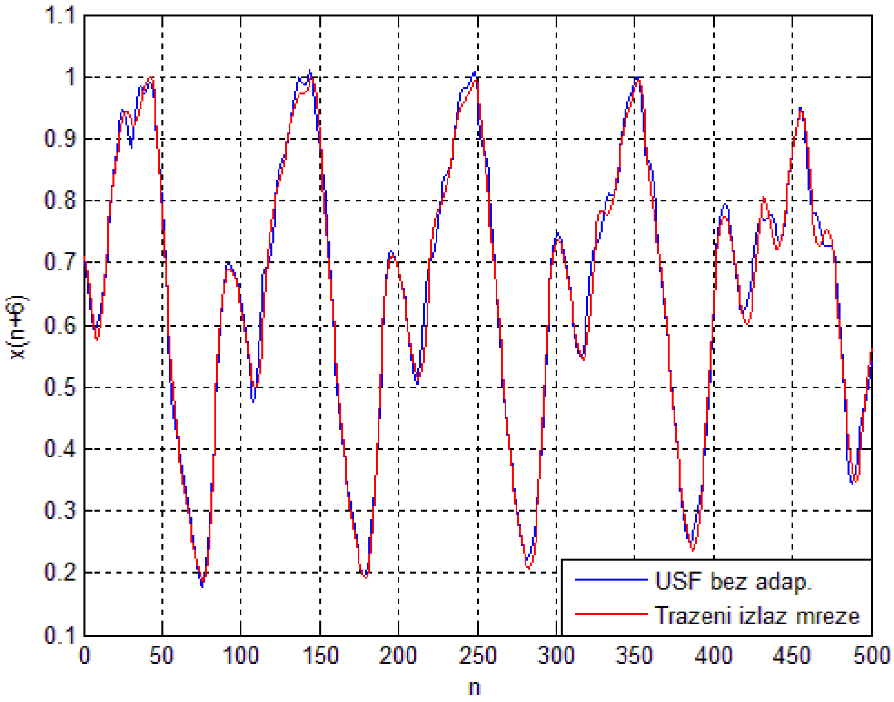



**4.7. Odaziv mreže s USF aktivacijskom funkcijom sa adaptacijom nagiba za test 2**

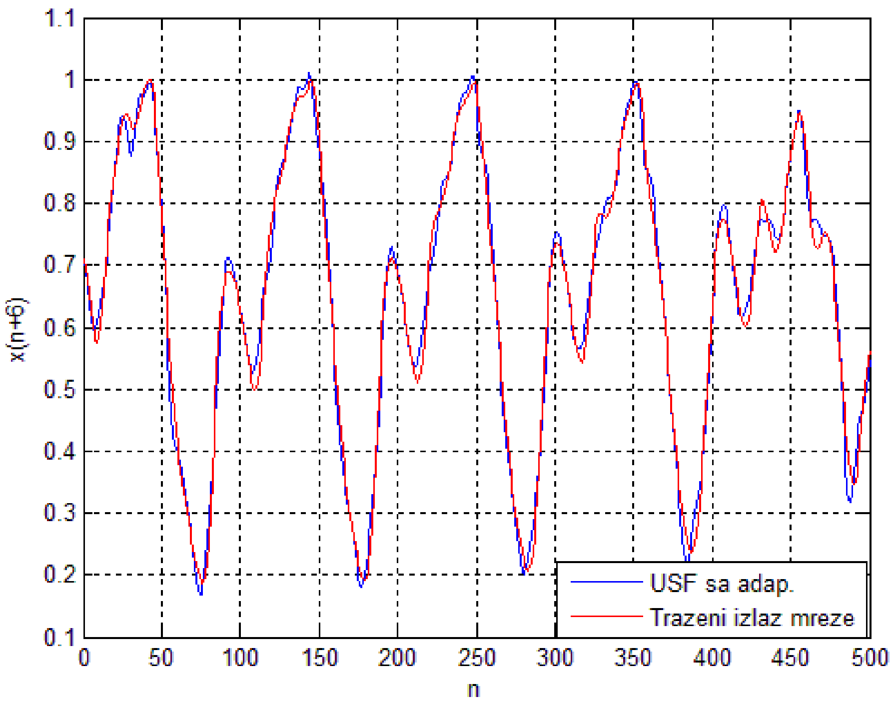



**4.8. Odaziv mreže s Gaussovom aktivacijskom funkcijom za test 2**

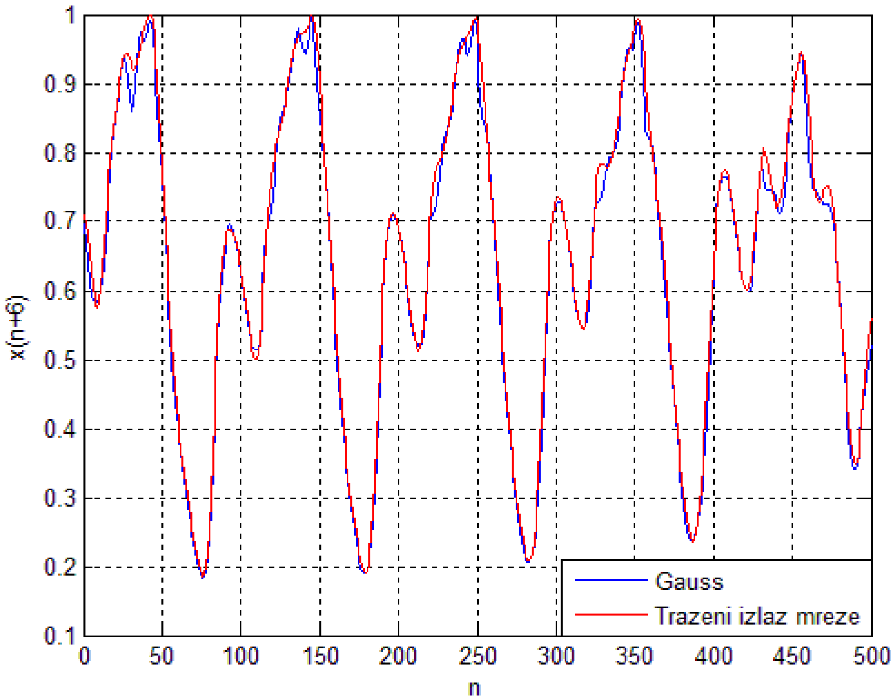
### 4.2.3 Test 3




**4.9. Odaziv mreže s USF aktivacijskom funkcijom bez adaptacije nagiba za test 3**




**4.10. Odaziv mreže s USF aktivacijskom funkcijom sa adaptacijom nagiba za test 3**




**4.11. Odaziv mreže s Gaussovom aktivacijskom funkcijom za test 3**

# 5. Zaključak

Cilj ovoga rada bio je analizirati primjenu tri različite aktivacijske funkcije skrivenog sloja statičke unaprijedne neuronske mreže. Radilo se o unipolarnoj sigmoidalnoj aktivacijskoj funkciji sa i bez adaptacije nagiba i Gaussovoj radijalnoj baznoj funkciji s adaptibilnim parametrima širine funkcije i pozicije centra funkcije.

Problemi odabrani za testiranje su bili jedni od najpoznatijih i najčešće upotrebljavanih u literaturi; XOR problem, poznatiji i kao problem linearno neseparabilnih uzoraka, te problem predikcije ponašanja nelinearnog kaotičnog sustava, gdje je korištena Glass-Mackeyeva jednadžba kaotičnog sustava.

Rješavanjem oba problema pokazala su se dva važna zaključka.

**Prvo**, mreža koja uči mijenjajući više parametara ima veću sposobnost generalizacije, kao i bržu konvergenciju ka traženom izlazu. Neuronska mreža koja je posjedovala Gaussovu aktivacijsku funkciju s neuronima skrivenog sloja sa dva adaptibilna parametra (naravno, uz težine skrivenog i izlaznog sloja) pokazivala je upravo takve prednosti nad mrežom s unipolarnom sigmoidalnom aktivacijskom funkcijom s adaptibilnim nagibom (jedan parametar vezan uz funkciju, te težine skrivenog i izlaznog sloja), dok je ona pokazivala prednost nad mrežom s unipolarnom sigmoidalnom aktivacijskom funkcijom koja nije mogla mijenjati niti jedan parametar funkcije skrivenog sloja.

Brza konvergencija uz korištenje Gaussove aktivacijske funkcije je bila posebno očita prilikom rješavanja XOR problema, kada je takva mreža u samo dvadesetak koraka smanjila NRMS na vrijednosti koju je mreža s USF s adaptacijom nagiba postigla tek oko šestotog koraka, a mreža s USF bez adaptacije nagiba nije to mogla postići niti u posljednjem, tisućitom, koraku.

Veća sposobnost generalizacije postala je vidljiva prilikom rješavanja drugog problema, predikcije ponašanja nelinearnog kaotičnog sustava, gdje su mreže bile ispitane na tri skupine podataka koje nisu učile. Iako je tamo brzina konvergencije bila manje-više slična za sve tri mreže, mreža s Gaussovom aktivacijskom funkcijom je radila daleko bolje predikcije za sva tri testa.

Dodatni dokaz prednosti posjedovanja više adaptibilnih parametara učenja se vidi iz rezultata mreža sa unipolarnim sigmoidalnim aktivacijskim funkcijama. Njihova je krivulja konvergencije bila vrlo slična, te je mreža bez sposobnosti adaptacije nagiba čak i u konačnici ostvarila marginalno bolji NRMS, no kada su bile podvrgnute testovima koje nisu učile, mreža koja je mogla mijenjati parametar nagiba aktivacijske funkcije prilikom učenja je pokazala veću sposobnost generalizacije od druge.

**Drugi** bitan zaključak vezan je za potrebno vrijeme učenja mreže. Već je ranije navedeno da su sve tri mreže bile slične koliko je to god moguće, te su sve radile isti broj koraka učenja, no svejedno, prilikom pokretanja računalnih modela mogla se osjetiti drastična razlika između mreža u potrebnom vremenu da se odvije učenje. Rješavanje XOR problema je za sve tri mreže zanemarivo kratko – potrebno je procesorskog vremena reda veličine jedne sekunde za simulaciju na osrednjem komercionalnom osobnom računalu. No, prilikom rješavanja problema predikcije te razlike postaju neugodno očite – mreži koja posjeduje Gaussovu aktivacijsku funkciju u neuronima skrivenog sloja treba četiri do pet puta više da obavi učenje nego ostalim dvjema mrežama, i uz to, mreža koja koristi unipolarnu sigmoidalnu funkciju s promjenjivim nagibom treba dvadesetak posto više vremena da obavi učenje od mreže koja nema mogućnost promjene parametra. Kao odgovor tom problemu može se predložiti modifikacija algoritma učenja neuronske mreže dodavanjem momentuma, odnosno zamaha u izraze za promjenu parametara učenja mreže s obzirom da se tom modifikacijom može ubrzati učenje i do deset puta. Također bi se dalo razmisliti o korištenju drugog algoritma učenja, kao i o korištenju EBP algoritma gdje se učenje odvija po skupu umjesto po uzorku. Tako bi se parametri učenja mreže mijenjali samo jednom – nakon prolaska cijelog skupa za učenje kroz mrežu, što bi teoretski trebalo ubrzati vrijeme učenja za probleme s velikim skupovima (XOR je ima 4 ulazno-izlazna para, ali u predikciji nelinearnog kaotičnog sustava imali smo ih 500!) u odnosu na metodu učenja po uzorku koja parametre učenja mijenja prilikom prolaska svakog ulazno-izlaznog para za učenje.

# Literatura

[1] D. Majetić, B. Novaković, M. Široki, *Umjetne neuronske mreže*, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2011.

[2] D. Majstorović, *Završni rad*, Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2012

[3] D. Majetić, *Neuronske mreže – podloge za predavanja,* Fakultet strojarstva i brodogradnje Sveučilišta u Zagrebu, 2011.